In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 1, lines 28 to 32 as follows:

--An 8-phase scaler restricts the resultant point to one of eight positions (i.e. eight sets of weighting values for each of the source ~~points)between~~ points) between any two adjacent source points. Fig.2 drawing demonstrates an 8-phase scaler. The points {p,q,r,s,t,u,v,w} represent the eight resultant positions that are the only resultant points that may exist between source points {w-1, w}--

Rewrite the paragraph at page 5, lines 14 to 27 as follows:

--The drawing of Fig. 4 and Fig. 5 shows the theory being applied to a 4-8-QSinc co-efficient curve with two different approximation curves. Each of Fig. 4 and 5 is using a different approximation curve (the solid line curves) to approximation the QSinc co-efficient curve. The approximation fits the phase curve as close as possible to the true curve and maintains a value of one when you add up the coefficients. The dotted line superimposed on the drawings is the true Sinc curve. Both curves were verified by software simulation to yield very high quality output images that were approaching true Sinc(x) quality. The arguments and reasoning used to derive the values of the elements in the co-efficient tables to the left of each of the two curves are the same as those used to derive the values of the co-efficient table described in the section titled "Linear ~~Scaler "~~ Scaler" above. The only difference is that a different normalization value is used. Instead of normalizing to 64, we normalize to ~~8(8~~ 8 (8 points to the next ~~source) ,~~ source), restricting the values of {*a,b,c,d*} to be in the range ~~(maintain~~ (maintain a value of one when you add up the coefficients):--

Rewrite the paragraph at page 7, line 24 to page 8, line 2 as follows:

--For each output pixel, the step size is used to determine how many input pixels to skip over. For each output pixel, the fractional value is added to an accumulator. If the accumulator rolls over (becomes > 1), an extra input pixel is skipped. This accumulator is masked to produce the phase value in the desired range (0-7 for 8 phase scalers). The phase value is used to index into a co-efficient table and retrieve the four values for a, b, c and d in the following "new pixel" formula. The values of A, B, C and D are the closest four input ~~pixel~~ pixels relative to the step position of the output pixels (four horizontal pixels for x stepping and four vertical pixels for y stepping). (a+b+c+d) is the normalization ~~(a~~ (a value indicating how many bit positions to shift the accumulated value to the right) value.--

Rewrite the paragraph at page 8, line 29 to page 9, line 14 as follows:

--Referring to Fig. 6 there is illustrated a low cost 4-8Qsinc Xscaler according to one embodiment of the present invention. The timing is illustrated in Fig. 7. A serial stream of source data (SData) is walked through a series of latches (A, B, C, and D) with a source clock (SClk). The content of the data in each latch (A,B,C, and D) corresponds to the variable of the same notation (A,B,C,D) in the formulas above. Two select lines (SSelect[1,0]) present the content of the four latches (A, B , C and D) one at a time to a limited fixed multiplier. At the same time, the appropriate co-efficient code (a, b, c, or d) is presented to the multiplier. The multiplier comprises the shifts and adds in the first adder. The shift >>1 is a divide by 2, the shift >>2 is a divide by 4 and the shift >>3 is a divide by 8. The first adder is

illustrated in Figs. 6A and 6B. The input N in Fig. 6A and 6B represents the input from each of the latches A,B,C or D The input N shifted by 3 (>>3 or divide by 8) equals N1,the input N shifted by 2 (>>2) or divide by 4) equals N2 and the input N shifted by ~~1(>>1~~ 1 (>>1 or divide by 2) equals N3. The coefficient code n is provided by lines X,Y and Z in Fig.6A provide either a 1 or 0 at the lines X,Y and Z such that the input at N1 for example is multiplied by a 1 or 0, N2 is multiplied by a 1 or 0 and N3 is multiplied by a 1 or 0. The outputs N1 = N1 or 0, N2= N2 or 0 and N3 = N3 or 0 are added to produce N*n for the example. This process is repeated for the four latched inputs of A,B,C, and D to give the products A*a,B*b,C*c, and D*d. The first three products are latched into latches A*a, B*b, and C*c. After the fourth source latch (D) is presented to the multiplier, and it's product is at the multipliers output, the content of the three latches (A*a, B*b, and C*c) are added together with the product of the D*d operation. The sum of these four products is the output or destination (DData) data.--

Rewrite the paragraph at page 9, lines 17 to 19 as follows:

--1. Multiplexing circuitry to parallel load the first pixel of ~~very~~ every line into the four latches (A, B, C, and D) with just one clock cycle.

Rewrite the paragraph at page 9, line 19 as follows:

--2. Details about the two ~~adders..~~ adders.--

Rewrite the table at page 10, between lines 15 and 15 as follows:

--

4-8-Qsinc Co-efficient Table									
~~Decimal~~ <u>Decimal</u> view					Binary view				
Phase	a	b	c	d	Phase	a	b	c	d
0	0	1	0	0	0000 0000	0000 0000	1000 0000	0000 0000	0000 0000
1/8	0/8	7/8	1/8	0/8	0010 0000	~~1111 1111~~ <u>0000 0000</u>	0111 0000	0001 0000	~~1111 1111~~ <u>0000 0000</u>
2/8	- 1/8	7/8	3/8	- 1/8	0100 0000	1111 1111	0111 0000	0011 0000	1111 1111
3/8	- 1/8	6/8	4/8	- 1/8	0110 0000	1111 1111	0110 0000	0100 0000	1111 1111
4/8	- 1/8	5/8	5/8	- 1/8	1000 0000	1111 1111	0101 0000	0101 0000	1111 1111
5/8	- 1/8	4/8	6/8	- 1/8	1010 0000	1111 1111	0100 0000	0110 0000	1111 1111
6/8	- 1/8	3/8	7/8	- 1/8	1100 0000	1111 1111	0001 0000	0111 0000	1111 1111
7/8	0/8	1/8	7/8	0/8	1110 0000	0000 0000	0000 0000	0111 0000	0000 0000

--

Rewrite the paragraph at page 11, lines 11 to 28 as follows:

--The timing controls for a linear Scaler is illustrated in Fig. 9 and is composed of a small down counter (skip counter), a couple of flip-flops, a few gates, and a Phase Lock Loop (4xPLL) circuit for generating the MasterClock which has to run at four times the frequency of the incoming SourceClock. The SClk clock is synchronized to the SourceClock by the PLL. If the value on the IBus is greater than one, the next resultant pixel is not generated in the current SClk cycle. On every SClk cycle the value in the Skip Counter is decremented by one until the counter counts down to zero. When the Skip Counter has a value of zero, it allows the flip-flops to run through a four-count cycle. The DClk1, DClk2, DClk3, and DClk4 are generated from this four-count cycle. The SSelect [1,0] are also generated by this four-count cycle. When the Skip Counter has a non-zero value, the four-count cycle is stalled. The incoming data (SData) is shifted through the four input latches (A, B, C and D), but a new resultant data (DData) is not generated. In ~~Fig. 9~~ <u>Fig. 8</u> (X Co-efficient Generator), note that a new value may appear on the IBus at the end of each four count cycle (clocked by DClk4). Because this is a down scaler, the IBus will always be loaded with a non-zero value at this time (unless the IntStep

register was improperly programmed). This will reset the X Timing Controls (XTC) four count cycle mechanism. In the reset state, SSelect[1,0] will equal 00 and DClk1, DClk2, DClk3, and DClk4 will all be low.--

Rewrite the paragraph at page 11, lines 29 to 30 as follows:

--Not shown in the drawing of ~~Fig. 9(for~~ Fig. 9 (for brevity and because they are considered to be straight forward processes) are:--

Rewrite the paragraph at page 12, lines 2 to 16 as follows:

--The Scaling Pipe of Fig. 6 can be modified for speed as illustrated in Fig. 10 to run four parallel pipes from the four latches (A, B, C, and D) to the DData Out adder. This will eliminate the need for a PLL and will allow the Scaling Pipe to run at the Source Clock rate. X Timing controls is reduced to a Skip Counter and the two clocking signals SClk and DClk4. The column selection of the Co-efficient table is also eliminated and all four co-efficient values are output to the scaling pipe concurrently. The Co-efficient table outputs four co-efficient values per clock cycle (one for each of the four parallel pipes). The overall design becomes simpler, but will cost more. There will also no longer be a need for an external clock. All internal clocking will be derived from the Source Clock. This will most likely be the version of the X Scaler that most designs will end up using. For the extra cost you get both a faster scaler and a much more simplified one. The following drawing shows the modified X Scaling Pipe. The drawings of the Timing Controls or the Co-efficient Generator are not provided because the modifications to them is straight forward and easy to determine.--

Rewrite the paragraph at page 12, lines 18 to 21 as follows:

--1. ~~Aliened~~ Aligned to Pixel first pixel of the Source Data. If the first pixel of the source is latched into ~~al~~ all four latches (A, B, C, and D) on the first Source Clock of each line, it makes no difference which set of co-efficient (a, b, c, d) values are used. The First pixel out on Ddata for that line will equal the first pixel in at SData.--

Rewrite the paragraph at page 15, lines 3 to 15 as follows:

--In accordance with the present invention the steps are ~~1.)~~ 1) generate a quantized table for any continuous coefficient curve using the four rules of ~~1)The~~ 1) The sample data must be consistent. If it's video or graphic data, all input pixels have to be of the same size and equally spaced. In the case of audio data, all samples have to be taken with a consistent sampling clock. Other forms of data have to be consistent in their appropriate domains as well. ~~2.)The~~ 2) The quantization points must be consistent. For example an 8-phase scaler has eight quantization points between each and every adjacent pair of pixels. ~~3)The~~ 3) The co-efficient curve must be symmetric about its centerline. ~~4.)The~~ 4) The quantization values must be selected so that the sum of all the co-efficient values in the co-efficient curve over all the source points for a given resultant point is always equal to the normalization value. The normalization value selected for a given approximation curve will be driven by this rule. Apply to the table in hardware using a step counter to locate the coefficient in the table. Apply the coefficient to a scaling pipeline to get the scaled data.--

Rewrite the paragraph at page 15, lines 16 to 19 as follows:

--While in the above description the curve approximated is for QsincX scaler any type of continuous nonlinear curve can be

approximated in a similar manner. For a VGA scalar the coefficient curve may be $Cos^2[(Pi)/2]$. Various ~~modifications ,alternative constructions ,and~~ modifications, alternative constructions, and equivalents may be used within the scope of this invention. --